



Microsoft Imagine

Welcome to the Microsoft Imagine Coding Club Starter Kit!

[Microsoft Imagine](#) provides all of the tools necessary for anyone to teach kids how to code. Yes, anyone! The resources in this coding club kit give you everything you need to lead an engaging coding camp, regardless of your previous experience with coding.

Why Coding?

Education is shifting to prepare kids for the future job market, and code literacy is just as essential to this generation as math, science, and languages. Coding encourages creative thinking, motivates problem solving, and builds confidence and perseverance in the face of challenges. Not only does coding build competency, but it also provides useful crossover skills in other disciplines outside computer science and programming. And like solving any interesting puzzle, coding can be totally fun!

Yes, YOU Can Teach a Coding Camp!

The Microsoft Imagine [coding kits](#) are designed so that anyone with the desire to do so can teach a coding camp, even if you've never written a single line of code.

We provide kits that make camp planning easy. Kids are immersed into ready-made projects that let them quickly see tangible, meaningful results. You can do our kits in any order, and add layers of gaming context and additional challenges with our kit guides.

Your job as camp leader will be to facilitate and excite. You can learn to code alongside your campers. If you have motivated campers who surpass your coding knowledge, enlist their help for collaborative teaching moments among your campers.

Still feeling nervous? Review the basics in the appendices of this document and try a tutorial or two.

Coding Projects for Every Camp

With free introductory coding tools like [Kodu Game Lab](#), [Project Spark](#), and [TouchDevelop](#), Microsoft provides a diverse range of games and projects so you can find the right fit for your campers' ages, interests, and coding experience. The number and flexibility of the camp lessons lets you structure any length camp you prefer, from an hour of code or a one-day hackathon to a camp that meets on a regular basis for a school semester.

Kodu Game Lab lets you create online games with a simple, visual programming language that's great for younger students as well as for any beginner coder. The Kodu coding kits enable kids to code a missing piece of a mini game, thus completing the game.

1. [Minigolf](#): students will create controls for different types of putting shots.
2. [Kodu Kart](#): students will program different object pickups and their effects in a racing game.
3. [Air Delivery](#): students will program 3D controls and collision reactions in this hot air balloon navigation game.
4. [Air Hockey](#): students will code game scoring and a way to win the game.

Two more kits will be available soon:

5. **Kodu Defense**: students will add enemies and program timing elements in a tower defense-style game.
6. **KoduMan**: students will program player controls, scoring, and power-ups in a classic maze-style game.

Project Spark uses a similar visual programming language to make games for Windows 8 or Xbox One. Its intricate world-building tools and adventure game templates make it a great choice for slightly older kids (10 and up), or students that have a strong understanding of Kodu.

In the Project Spark lessons, students will focus on creating game mechanics in a wide variety of games.

1. [The Warrior's Crucible](#): students will code leveling up scenarios in an arena battle game.
2. [Predation or Starvation](#): students will create objects to keep goblins from raiding farmlands, and maintain a balance between the squirrel and goblin populations.
3. [Revive the Adventurer Alliance](#): students will program ways for teammates to revive in a 2D platform game.
4. [The Cost of Commerce](#): students will create and fix variables in a game where the player is a peddler selling goods in different towns.

Two more kits will be available soon:

5. **Lunchtime Blues:** students will program character conversations, and provide ways to interact with objects based on conversation flags.
6. **Plotting a Stealthy Path:** students learn about programming AI behaviors to detect sneaky players.

TouchDevelop uses a block-style programming language, letting you create a variety of multiplatform casual games or apps. This is a great coding tool for kids ages 10 and up, as well as for student who have prior experience with the Scratch coding language.

The TouchDevelop lessons walk students through creating a game or app from scratch.

1. **[Magic Ball](#):** Students work on creating a simple answer app and code randomized results when the device is shaken or swiped.
2. **[CompatiMatch](#):** students will code a simple algorithm to generate a couple's compatibility.
3. **[Brick Breaker](#):** students will set up sprites and collision in a physics-based game.
4. **[Doodle Guesser](#):** students will set up a topic generator and an empty canvas with drawing tools for this classic game.

This kit will be available soon:

5. **Endless Runner:** students will generate obstacles, set up gravity, and code a timer for an endless runner game.

You can teach these lessons in any order. Let the interest of your campers drive your lesson selection.

Location, Location, Location

You may be starting with an existing space, but if you don't already have a location in mind for your camp, check the many kid-friendly spaces in your community:

- **Schools:** Neighborhood schools with kids of coding age are a great place to start. Check access for after school, and see if weekends are an option.
- **Libraries:** Coding makes an excellent addition to the family programming many libraries do. As a bonus, many have computers available.
- **Community Centers:** Are there community centers or churches in your area that offer kids' programs?
- **Private Businesses:** Check bookstores, cafes, and kid play spaces as possible camp options.
- **Local organizations:** Look for existing kids groups like scouting troops or clubs.
- **Camping at Home:** There's always the informal option of camping at home. This works well if you're looking for an informal setting for camping with kids you know.

As you research spaces, be sure to check on these details:

- **Internet access:** Will all of your students be able to get online?
- **Outlets:** Is there room for everyone to recharge?
- **Desks or Tables:** Can all of the students sit comfortably at their computers?
- **Safety:** Is the location a safe and accessible one for kids?
- **Permits:** Are any permits required for your selected space?
- **Available hardware:** Does the location have computers that your students can use?

Computers for Camp

You can't code without computers! The good news is that you don't necessarily need a computer for every camper. [Pair Programming](#), where kids work together in pairs taking turns coding, works very well in a camp setting.

The devices you will need will vary with the coding tool you will be using:

Kodu Game Lab	Project Spark	TouchDevelop
Windows PC Xbox 360	Windows 8.1 PC Xbox One	Windows PC, Mac, or Linux Tablet Mobile Phone
System Requirements: Windows 7, Vista, or XP A graphics card that supports DirectX 9.0c and Shader Model 2.0 or higher .NET Framework 3.5 or higher XNA Framework 3.1 Redistributable	System Requirements: Windows 8, x86 or x64 CPU: Intel Core i5 or better RAM: 4 GB Hard Drive Space: 2 GB Video: Intel HD 4000 or better	System Requirements: <u>Desktop Computers:</u> Internet Explorer 10 Chrome 22+ for PCs and likely Macs, Linux Firefox 16+ for PCs and likely Macs, Linux <u>Tablet/Mobile Phone:</u> Windows Phone 7.5 iPhone/iPad iOS6 Safari Android Chrome v18+

In addition to devices, having a way of looking at projects as a group, though not critical, is very handy. Projectors or white boards work well to get everyone on the same page.

Be sure that all students will be able to access the Internet.

If you don't already have the necessary hardware to run your camp, consider these options:

- Check with the location. Schools and libraries may be able to fill your technology needs, particularly if you're providing this service for their students.
- Ask kids to provide their own. Many kids (or their parents) may have their own laptop or tablet to bring.
- Grants: Many groups see coding as an investment, and grants can be a way to foot your hardware bill.

If You Plan It, They Will Come

You have the location; you have the devices; now you need the campers! If you're not designing your camp with a particular set of campers in mind, there are many ways to fill your camp seats:

- Tap into the audience at your location. If you're in a school, send an email to parents. In a library or community center, try getting a plug on their website or in their newsletter. Or, hang flyers in visible locations. Just because your camp is high tech doesn't mean your marketing has to be.
- Check with local scout troops. A coding camp is a great way to earn a badge!
- Reach out to parent bloggers, kid-friendly websites, and newspapers in your area.
- Spread the word of your camp on parent email lists and on social media.
- Ask enrolled kids and parents to recruit friends.

Sample Email

Here's a sample email you can modify to announce your camp. Simply replace the text in [] with details applicable to your coding camp.

Enroll your kids in coding camp today!

Dear Parents,

If your kids love to play computer games, imagine their excitement when they start coding games themselves! In our coding camp, kids will code a wide variety of games like [air hockey, a brick breaker game and an endless runner] using [Kodu Game Lab, TouchDevelop and Project Spark]. They'll have a blast while learning important computer skills.

Who: Boys and girls [ages 9-12]

Where: [Camp Location]

When: [Every Wednesday from 3:00 – 4:00]

Kids will need to bring a [appropriate device for the tool you're using].

No prior coding experience necessary!

To register, contact [contact info].

Encouraging Coder Girls

Recent research shows that, on average, [only 25% of tech jobs are held by women](#). One way to close that gender divide is to encourage girls' interest in coding and help them to persevere once they start down that path. Here are four ways you can enroll and retain girls in your coding camp:

1. Make it explicit in your camp information that girls are welcome. Instead of saying the camp is for “kids ages 9 – 12,” for instance, say “boys and girls ages 9 – 12.” If your flyer shows a photograph of kids coding, make sure there is a girl in the picture.
2. If girls are far outnumbered by boys, be sure to offer words of encouragement to not drop out of the camp, just because they feel out of place.
3. During camp, show girls that they are welcome. From day one, make sure they have the space to participate without being talked over or dismissed.
4. Or consider hosting a girls-only camp.

Planning your Camp Sessions

Where to Begin?

Once you have your location, computers, and campers, you're ready to start planning your lessons. Remember, the lessons are flexible and can be done in any order. For your first activity, consider picking the one that you're most excited about – enthusiasm is contagious. For subsequent activities, consider letting your students decide what they'll learn next. Their involvement in decision-making can boost their engagement in the camp.

As you think about the number of students in your class, think about how many facilitators and devices you'll need. Depending on the age of your students, it's good to have a facilitator for every 10 students. One teacher can lead while the other facilitator(s) can answer questions and help students troubleshoot. Advanced students can make great facilitators. Having one computer for every two students works well. In a pair programming approach, one student can lead while the other one types, then they reverse.

How Long is a Lesson?

Our coding kits is designed to last about 30 minutes from start to finish, but that can stretch to an hour or more as you factor in setup and cleanup, troubleshooting, and the additional context and challenge that you can add as the camp leader.

Following the session structure below, you can add tremendous value to the lessons. Facilitate conversations about the various aspects of game design. Encourage teamwork in code troubleshooting and debugging. If you're a gamer, bring your background knowledge about different types of games. Ratchet up the challenge when everyone has successfully completed the lesson using the remix ideas provided. You may even decide to work on a lesson over more than one camp session as your students have creative ways to remix the different projects.

Structure Your Session

For each camp session, follow these steps to introduce and provide context for the coding challenge, lead discussions, and challenge students to deepen their understanding of the code.

1. **What will you be coding?** Use the lesson description to talk to your students about the game you'll be working on. Invite students to think of games they've played that work in a similar way. Ask about the features that make that game fun.
2. **What's the activity goal?** Describe to your students what they're going to be coding, and invite discussion on why that part of the code is important to the activity. For instance, what's a sports game without scoring?
3. **Play the sample game as it exists in the start of the lesson.** If you have a projector or whiteboard you can do this as a class, otherwise you can have students to this at their computers. Discuss the moments in the game where the goal will be inserted and the effect it will have on the quality and fun of the game.
4. **Code independently.** When everyone is clear on the goal of the game, have students progress at their own pace through the step by step video instructions in the lesson. With a projector or whiteboard, you can play these videos for everyone at once. Otherwise, have students work with headphones at their computers as noise will be a factor. While coding is happening, be sure to give students space to troubleshoot on their own, or with their peers, before jumping in.
5. **Regroup for further discussion.** When students have completed coding the lesson, bring attention back as a group. Play the resulting game and talk about how the new code improved the game. Ask about challenges the students faced as part of the coding. End by brainstorming ways of pushing the game even further. How might students expand on the coding they just completed? What other features could be added?

6. **Remix and challenge!** Each lesson comes with ideas to remix and change the games as further coding challenge. You can also use your discussion as a jumping off point to provide students with further coding challenges.

Camp Planning Checklist

Planning

- Find location
- Pick day(s) and time(s)
- Recruit students
- Recruit facilitators or advanced students (optional, depending on class size)
- Find device resources (if not student required)

Week Before

- Test and charge computers/devices
- Determine need for any power strips or extension cords
- Load any necessary software/plugins
- [Try activities ahead of time](#)
- Send confirmation email to students, and have them set up accounts for the coding tool(s) you'll be using

During a Session

- Pair up students to work together
- Use session planning tips to invite discussion
- Ensure that all student have equal opportunity to code and participate in discussions
- Invite students to share their work with the camp
- Make sure students properly power down devices at the end
- Take pictures to share with parents

After a Session

- Email parents to thank them for participating, and encourage them to have their child show them some completed projects
- Send camp photos to parents and students

Facilitate Troubleshooting

When running your camp, remember: You don't have to have all the answers all the time. Part of the learning process for your students is in the troubleshooting and debugging.

There are a few ways you can work with students to troubleshoot. If working individually doesn't work for your group, have them work in pairs or small groups. Encourage students

to talk through possible solutions to the problems they encounter. You can also bring questions, bugs, or problems a student encounters to the whole class to discuss and troubleshoot.

As you teach your camp, you may encounter motivated students that surpass your knowledge of the coding tool. Fear not! These students can be great allies to help the rest of your camp.

You can also head online for help with these resources:

- [Kodu Game Lab Community](#)
 - [Kodu Support](#)
 - [Kodu Classroom Kit for Educators](#)
 - [TouchDevelop Teachers Group](#)
 - [TouchDevelop Blog](#)
 - [TouchDevelop Forums](#)
 - [Project Spark Guides](#)
 - [Project Spark Support Forum](#)
 - [@XBoxSupport on Twitter](#)
-

Appendix 1: Kodu Game Lab Basics

Kodu Game Lab gives kids the power to create games on the PC or Xbox 360 using a visual programming language that is easy to learn. Once kids have mastered the basic building blocks of game design, they can unleash their creativity to create fun, customized game experiences.

Kodu is available [for download on PC](#), or from the Xbox Marketplace in the Indie Games channel.

Kodu Game Lab Basics

When you program in Kodu, what you're really doing is describing how objects interact in a world. Objects can mean the character you're controlling, obstacles in your character's path, objects to be collected, other characters to have a conversation with, and more. The world refers to the space where your game is played, from an open grassy field, to an air hockey game table, to a maze, to a fantasy world filled with rolling hills and elaborate castles.

When you program objects in Kodu, you get them to **do** things in your game. This is exactly how the programming interface is presented: **When, Do**.

The "When" can mean a player interaction, such as when a player presses an arrow key on the keyboard. It can also mean two game objects interacting with each other, like when they bump into one another. Or, "When" can refer to specific points in the game, such as when the player achieves a certain score or a timer runs out of time.

The "Do" is what you want to have happen for each "When" scenario. "When" an arrow key is pressed, the "Do" command can tell the game to move the character in the direction of the arrow. "When" an object is collected, "Do" increase the score by 10 points. "When" the timer runs out, "Do" end the game, and so on.

You'll soon see how you can layer these commands to build an exciting game. Assign several different Do commands for different When commands. Use the plus sign after the different commands to define further details. Do you want the object to move quickly or slowly? Do you want to move along a path or wander all over? All of these commands are at your fingertips.

Navigating Kodu

Follow the onscreen commands to see how to select and use tools and navigate through the game. Whether you're using the keyboard and mouse or an Xbox 360 controller, these commands update as you select different tools.

Home Menu: This is where you can save your world, load new worlds, and exit to the Main Menu.

Play Game: Try playing your game at any time using the play button. It's important to play often as you code to spot any trouble areas and to see what's working well. Press Esc (Back) to return to editing mode.

Move Camera: Use the hand to change your view of your world. You can zoom in and out of your world using your scroll wheel (On Xbox, use L and R to zoom, right stick to tilt, and left stick to move).

Object Tool: Here is where you'll find all of the objects you'll need for your game. Click (press A) on the Object Tool then click on your terrain to bring up a submenu. [Show submenu] Once you select an object from the menu, it will appear in your game. Roll over the object to change its color, and right click (press Y) to program.

Path Tool: Use this tool to create paths for your objects to move along (On Xbox, paths are available under Edit tools).

Ground Brush: Paint, add, or delete ground using this brush. Use the submenus to change the ground pattern or the shape of the brush.

Up/down, Flatten, and Roughen: Use these tools to modify your terrain, creating smooth areas, hills and mountains, and rocky terrain.

Water Tool: Use the water tool over any terrain to submerge it underwater. Change the look of your water using the submenu.

Delete Tool: Use this tool to remove objects from your world.

Change World Settings: After you've mastered the basics, use these settings to modify the details of your game. Here you'll also find debugging and volume tools.

Getting Stared with Tutorials

Launch Kodu on your PC or Xbox. From the Main Menu, select "Load World." Many of the choices are labeled as tutorials. You can start with the "First Tutorial" or dive right into the Maze Tutorial.

Note that when you load worlds, they are in play mode by default. Press Esc to enter the editing mode (Back button on an Xbox controller).

For further tutorials and resources, visit [Kodu Game Lab](#).

Appendix 2: Project Spark Basics

Project Spark is like a super-charged version of Kodu, letting you create professional-looking games with similar easy-to-use terrain tools and visual programming language.

Project Spark is available for [Windows 8.1](#) and [Xbox One](#).

Project Spark Basics

Project Spark has sophisticated terrain-building tools that let you sculpt and paint amazing landscapes for adventure games. Biome tools let you instantly create natural elements like rivers, and plants and trees generate automatically as you paint your natural landscape.

Character tools let you make both first-person and third-person styles games, allowing for a huge variety of games you can make with Project Spark.

Project Spark works with the same style of When/Do code as Kodu. You set up the code in the same way. “When” the player does something, or objects react to one another in the game, you code what you want the game to “Do.”

The basic character in Project Spark is preprogrammed with a large amount of code to let it instantly interact in your level. You can leave this code as is, or start experimenting and customizing it as you code your game.

Navigating Project Spark

When you enter the Create mode of Project Spark, you have the option to continue work on your levels in progress, start with an empty world, use the Learn to Create option to start with a game template, or view tutorial videos.

Once you’re in the level editing mode, use your keyboard and mouse to get around. Use the WASD keys on your keyboard to move through the game space. The E and Q keys will take you up and down in space. Right clicking the mouse will let you rotate around your cursor. If you get lost in your space, use the “Center on Player” or “Camera Top Down” buttons on the right side of the screen.

If you have an Xbox controller, you can also use it to navigate through the game world. Use the left and right sticks to glide the camera through the game space. The X and Y buttons will move the camera up and down vertically.

Next, there are the tools that you’ll use to create your game:

Play: [Click here to play your game.](#)

Biome: Here you can select the look of your game. This lets you paint things like trees and rivers right into your game.

Paint: Select looks from the Material Palette, along with the size and intensity of your paintbrush, to paint a wide variety of textures onto your world.

Sculpt: Use this tool to shape your environment. You can add using sculpt, remove using erode, and further fine tune using the different options.

Prop: This lets you control the objects and characters in your game. Click the brain icon to access the When/Do code for that object or character.

When you pause gameplay using the ESC key, you'll bring up a pause menu where you can continue editing, resume, or restart your game. There is also a submenu of World Settings that let you further change the look of your world with different lighting, sun angles, time of day, etc. [Show Pause menu]

Getting started with tutorials

From the main menu, click the Create button to begin. On the next screen you'll see a button labeled "Tutorials." You can also find [tutorials on the Project Spark website](#).

Appendix 3: TouchDevelop Basics

TouchDevelop lets kids create their own apps on a phone, tablet, or laptop. Blocks with code pieces make coding easy. TouchDevelop has three skill levels, beginner, coder, and expert, making it a great, adaptable tool for different kinds of coding camps.

If you have any experience with Scratch, you'll find that TouchDevelop is similar. In fact, you can even program games using a Scratch code library.

[TouchDevelop](#) works on Windows, Mac, iOS, Android, and Linux.

TouchDevelop Basics

When coding using TouchDevelop, you're creating scripts. These scripts are a series of block-based code that run your game. TouchDevelop is filled with libraries of code, art, and sounds, and you also have the capability to upload your own art and sound.

What makes TouchDevelop great as you're beginning to learn to code are game templates. When you launch TouchDevelop, you'll see a button to "Create Script." That brings up a

submenu where you can launch a blank script to start a new game from scratch, or choose a blank game template to start with the building blocks already in place for a great game.

When you begin a game in Beginner mode, you'll see a purple block labeled "function main ()," and within this block you'll create a series of blocks that make your game work.

Navigating TouchDevelop

Once you have selected a script from the main hub, you'll be on a split screen that shows the code at the left and a game screen on the right.

These buttons will help you navigate:

My scripts: click here to navigate back to all of your scripts, including tutorials in progress.

Run: Click here to play your game.

Undo: Click here to undo your previous actions. You can keep clicking to work backwards through your coding history.

Split: Click here to bring up more options, such as adding pictures or sounds.

When you're in the "My Scripts" view, you'll see a list of all of your scripts in progress. Click on a script to see more options, including:

Edit: Click here to return to the editing view.

Run: Click here to play your game.

Publish: Click here when your game is done and you're ready to share it.

To navigate the code blocks, click on a green block. That will bring up a menu of choices for the selected block. Click on different parts of the text in a green block to bring up different choices. You can also type into areas of the green blocks to change variables.

To add new blocks, simply click on a green block and then click the + button either above or below it. You can also rearrange blocks by clicking and dragging them to a different position.

Getting started with tutorials

On the TouchDevelop hub, you'll see a large section labeled "Tutorials." Click on the ["All Tutorials"](#) button to see all that are available, from game tutorials to social apps to advanced algorithm tutorials. The tutorials labeled "Beginners," especially the ["Jetpack Jumper"](#) tutorial, are a great place to start.